U-DWM-Taller-3 - Fleet Management System

Quick Start

Prerequisites

- Node.js 18+
- MongoDB 5+ (or Docker)
- npm

Installation

```
# Install dependencies
cd backend && npm install
cd ../frontend && npm install
```

Running Locally

Terminal 1 - Backend:

```
cd backend
npm run start:dev
```

Terminal 2 - Frontend:

```
cd frontend
npm run dev
```

Terminal 3 - Database (if needed):

```
docker-compose up -d
```

Seed Data

```
# Create 20 vehicles with faker
npm run faker:create

# Distribute updates across the week
npm run faker:update
```

```
# Or use static seed
npm run seed
```

Project Structure

- **backend** - NestJS API with MongoDB
- **frontend** - React + Redux + Recharts dashboard
- **docker-compose.yml** - MongoDB setup

Key Endpoints

- `GET /fleet/dashboard` - Get dashboard data
- `POST /fleet/vehicle` - Create vehicle
- `PATCH /fleet/vehicle/:id` - Update vehicle
- `POST /fleet/seed` - Seed database

Features

- Real-time fleet tracking with KM history
- Weekly dashboard metrics (Km, Fuel, Active Fleet, Alerts)
- Historical data visualization
- Responsive mobile design
- Faker-based test data generation